BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

CORRECTED VERSION*

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 15/16, 7/00, 13/38

A1

(11) International Publication Number: WO 00/23905

(43) International Publication Date: 27 April 2000 (27.04.00)

(21) International Application Number: PCT/US99/24919

(22) International Filing Date: 21 October 1999 (21.10.99)

(30) Priority Data:
09/177,444 21 October 1998 (21.10.98) US

(71)(72) Applicant and Inventor: PORTER, Swain, W. [US/US]; 12511 89th Court, N.E., Kirkland, WA 98034 (US).

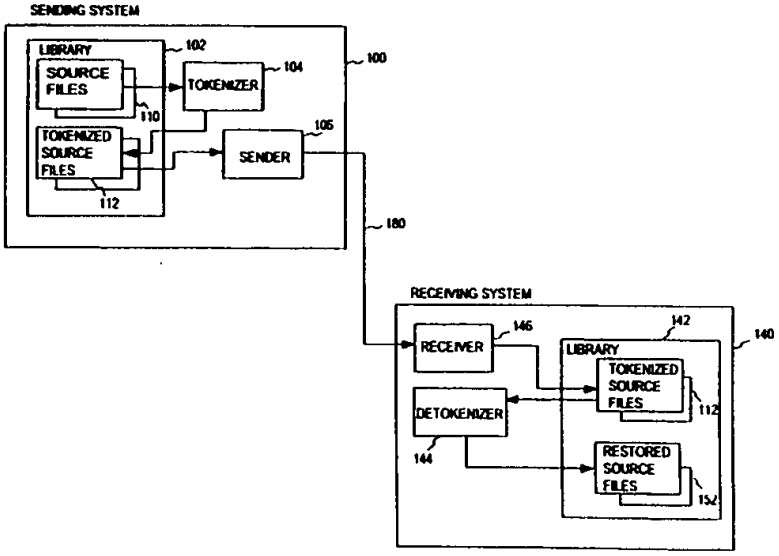
(74) Agents: AUYEUNG, Aloysius, T., C. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025–1026 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: TOKEN BASED SOURCE FILE COMPRESSION/DECOMPRESSION AND ITS APPLICATION



(57) Abstract

A source file (110) is transformed into a tokenized form (112) with at least language elements present in the source file (110) in an original form being substituted with corresponding tokens to reduce transmission bandwidth required to provide the source file (152) to another computer system (140). In one embodiment, operands present in the source file (110) in the original form are also substituted with corresponding tokens, and entries mapping the operand substituting tokens to the operands are maintained in a symbol table. The symbol table is also provided to the other computer system (140). In one embodiment, the computer system is a web server (100), and the source file (110) is a web page. The other computer system is a client computer system (140) requesting the web page from the web server. The requested web page is provided by the web server (100) to the client computer system (140) in the tokenized form, and the client computer system (140) is equipped with a browser having been enhanced with the ability to restore the provided web page to its original form.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Token Based Source File Compression/Decompression And Its Application

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of computer systems. More specifically, the present invention relates to methods and apparatuses associated with the distribution or provision of source files to other computer systems.

2. Background Information

With the recent advances in microprocessor, telecommunication and networking technology, increasing number of computer systems are being networked together. In turn, increasing number of situations arise where source files have to be transferred from one computer system to another computer system. The term "source files" as used herein includes but not limited to compilable or interpretable source files written in machine programming languages such as C, C++, HTML, XML, JAVA™, JAVAScript, and so forth. For example, everyday, millions of users are connected to the Internet downloading web pages from a multitude of web sites. Similarly, millions of users are doing the same within thousands of "corporate" intranets. In a new user centric software distribution paradigm, disclosed in co-pending U.S. patent application number <to be inserted>, filed contemporaneously, and entitled "User centric source control", it is envisioned that software products are distributed to client systems or their proxies in source form. All these activities further exacerbate the well known bandwidth problem confronting private as well as public networks. (Note that the "user centric" approach to source control contributes to the bandwidth problem only in the sense that the

approach is expected to increase the demand for source file transmission. For a given set of source files, its delta feature actually reduces bandwidth demand for maintaining and updating the set of source files.)

Various compression/decompression techniques are known in the art in the data or link layer to reduce the amount of data that needs to be transmitted from a sender to a receiver. For examples, a dictionary based approach replacing previously transmitted character string (e.g. "this string has been sent before") with a code is often employed in modem communication; the run length encoding approach (e.g. encoding a series of 10 0-bit as [0, 10]) is often employed in video signal compression, and a code based approach supplying the identity of a linear excitation code vector is often employed in audio compression. However, notwithstanding these multitude of data or link layer compression/decompression techniques available, as evident by the amount of research and development going into Quality of Service, Bandwidth Reservation, Virtual Private Network, and so forth, the problem of bandwidth in private as well as public networks is expected to remain with the computer and communication industry for years to come.

Thus, further improvement or contribution to alleviating the bandwidth problem, in particular, improvement that further advances the connectivity and exchange of information between computer systems, is desired.

SUMMARY OF THE INVENTION

In accordance with the present invention, a source file is provided from one computer system to another in a tokenized form to reduce transmission bandwidth requirement. In the tokenized form, at least language elements present in the source file in its original form are substituted with corresponding tokens. In one embodiment, operands present in th source file in the original form are also substituted with corresponding tokens, and entries mapping the operand substituting tokens to the operands are maintained in a

symbol table. In this case, the symbol table is also provided to the other computer system.

In one embodiment, the source file is also in either a base or delta form. A new entry is created for the symbol table whenever a new operand is encountered and substituted. In this case, subsequent to the initial provision of the symbol table, new entries associated with a base/delta source file are also provided to the other computer system to update the previously provided symbol table.

In one embodiment, the base/delta source files also have associated versioning control information. In this case, the versioning control information is also provided to the other computer system.

In one embodiment, the computer system is a web server, and the source file is a web page. The other computer system is a client computer system requesting the web page from the web server. The requested web page is provided by the web server to the client computer system in the tokenized form, and the client computer system is equipped with a browser having been enhanced with the ability to restore the provided web page to its original form. In one embodiment, the web page is also in a base/delta form having associated versioning control information, and the client computer system' browser is enhanced with the ability to reconstitute the requested web page using the associated versioning control information.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figures 1a-1c illustrate three exemplary embodiments of the token based compressed source file transmission of the present invention;

Figures 2a-2b illustrate one exemplary embodiment each for a collection of tokens and an associated symbol table suitable for use to practice the present invention;

Figure 3a-3b illustrate one exemplary embodiment each of the sender and the receiver's method steps in accordance to the present invention;

Figure 4 illustrates one exemplary application of the present invention to web servers and client systems accessing web servers; and

Figure 5 illustrates one embodiment of an exemplary computer system suitable for use as either a sender or a receiver system to practice the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in terms of operations performed by a computer system, using terms such as tables, files, data and the like, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and

otherwise manipulated through mechanical and electrical components of a digital system; and the term digital system include general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps performed in turn in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent, in particular, the order the steps are presented.

Referring now to **Figures 1a-1c**, wherein three exemplary embodiments of the present invention are shown. These three exemplary embodiments will be described in turn, referencing also **Figures 2a-2b**. First, referring to **Fig. 1a**, exemplary sending and receiving systems **100** and **140** are illustrated as being coupled to one another via communication medium **180**. More importantly, in accordance with the present invention, sending system **100** advantageously provides source files to receiving system **140** in a tokenized form, generated from an original form, thereby reducing transmission bandwidth requirement on communication medium **180**. For the illustrated embodiment, sending system **100** includes library **102**, tokenizer **104** and sender **106**, whereas receiving system **140** includes library **142**, de-tokenizer **144** and receiver **146**.

Over in sending system **100**, library **102** is used to store source files **110** in the original form and source files **112** in the tokenized form. Tokenizer **104** is used to transform source files **110** in the original form to source file **112** in the tokenized form. For the illustrated embodiment, tokenizer **104** effectuates the transformation by substituting language elements, such as arithmetic operators, relational operators and so forth, with tokens. The term "token" as used herein is intended to have similar meaning as the term is

commonly used by those skilled in the art of compiler technology, which typically includes a token class designation, e.g. arithmetic operator, and a class value, which may be a value designating a particular operator of the class. e.g. the value designating the "+" operator (see **Fig. 2a**, wherein an exemplary collection of language element substituting tokens is illustrated). Sender **106** is used to send source files **112** in the tokenized form to receiving system **140**, as described earlier. Sender **106** may send source files **112** in the tokenized form to receiving system **140** at its own initiative, at the request of a local requestor (not shown), e.g. an application, or a remote requestor (also not shown), e.g. an application on receiving system **140**.

In one embodiment, the programming language a particular source file **110/112** is written in is inferred from the file name of the source file, e.g. the file name including a file extension, such as "htm" for the hypertext markup language (HTML). In another embodiment, the programming language a particular source file **110/112** is written in is determinable from the properties associated with the particular source file **110/112**, which is integrally provided along with a particular source file **112**, when the particular source file **112** in tokenized form is provided to receiving system **140**. In yet another embodiment, sender **106** informs receiver **146**, the programming language the particular source file **110/112** is written in.

Still referring to **Fig. 1a**, over in receiving system **140**, receiver **146** is used to receive source files **112** in the tokenized form from sending system **100**, including as described earlier, its programming language, either integrally or particularly. Library **142** is used to store received source files **112** in the tokenized form, as well as restored source files **152** in the original form. De-tokenizer **144** is used to restore source files **112** in the tokenized form to source file **152** in the original form. For the illustrated embodiment, de-tokenizer **144** effectuates the transformation by restoring language element substituting tokens with their corresponding language elements. De-tokeniz r

144 is equipped with the language element substituting token to language element mappings for a number of programming languages. In one embodiment, source files **110/112/115** may be written in include but not limited to C, C++, HTML, XML, Java™, and JavaScript, and de-tokenizer **144** is accordingly equipped to handle the supported programming languages.

Communication medium **180** is intended to represent a broad range of communication medium known in the art, from local area networks (ethernet, token ring, etc.) to wide area networks (ATM, frame relay, and so forth). Accordingly, communication medium **180** will not be further described. Libraries **102** and **142**, sender/receiver **106** and **146**, as well as tokenizer/de-tokenizer **104** and **144** are also intended to represent a broad range of these elements known in the art. Thus, except for the manner these elements are employed to practice the present invention, individually, these elements will also not be further described.

Figure 1b illustrates an alternate embodiment. In this embodiment, sending system **100'** also advantageously provides source files to receiving system **140'** in the tokenized form, thereby reducing the bandwidth requirement on communication medium **180'**. Sending system **100'** similarly includes library **102'**, tokenizer **104'** and sender **106'**, while receiving system **140'** similarly includes library **142'**, de-tokenizer **144'** and receiver **146'**. Each of these elements perform the same functions as the corresponding element described earlier for the embodiment of **Fig. 1a**. The key differences between these two embodiments are in the manner in which tokenizer **104'** transforms source files **110'** in the original form to source files **112'** in the tokenized form, and de-tokenzier **144'** restores source files **112'** in the tokenized form to source files **115'** in the original form.

More specifically, in addition to substituting language elements with corresponding tokens to reduce transmission bandwidth requirement, as

described earlier, tokenizer **104'** further substitutes operands present in source file **110'** in the original form with corresponding tokens. Additionally, tokenizer **104'** further creates and maintains a symbol table **114'** for each group of related source files, e.g. those to be compiled and linkedited together. In particular, tokenizer **104'** creates a mapping entry for symbol table **114'** for each new operand it encounters and substitutes with a new token. For the exemplary embodiment of tokens illustrated in **Fig. 2a**, the class value of the operand class token will be set to a pointer pointing to the appropriate mapping entry in the symbol table (see **Fig. 2b**, wherein an exemplary embodiment of a symbol table is illustrated).

Symbol tables **114'** are also provided to receiving system **140'** by sender **106'** of sending system **100'**. In like manner, receiver **146'** stores the received symbol tables **114'** in library **142'**, making them available to de-tokenizer **144'** when it restores source files **112'** in the tokenized form to source files **152'** in the original form. In other words, in addition to the language element substituting token to language mappings de-tokenizer **144'** is equipped with, de-tokenizer **144'** further uses the operand substituting token to operand mappings in symbol tables **114'** to effectuate the restoration.

Figure 1c illustrates yet another alternate embodiment. In this embodiment, sending system **100"** also advantageously provides source files to receiving system **140"** in the tokenized form, thereby reducing the bandwidth requirement on communication medium **180"**. Sending system **100"** similarly includes library **102"**, tokenizer **104"** and sender **106"**, while receiving system **140"** similarly includes library **142"**, de-tokenizer **144"** and receiver **146"**. Each of these elements perform the same functions as the corresponding element described earlier for the embodiments of **Fig. 1a-1b**. The key difference between this and the earlier embodiments is the fact that source files **110"/112"/115"** are also in either a base or delta form, having associated versioning control information **116"**. Accordingly, sender **106"** also

provides receiving system **140"** with new operand substituting token to operand mappings for symbol tables **114"**, whenever sender **106"** provides receiving system **140"** with a base/delta source file **112"** in the tokenized form involving new operand substituting tokens, not previously employed in base/delta source files **112"** earlier provided to receiving system **140"**. Additionally, for this embodiment, sender **106"** also provides versioning control information **116"** to receiving system **140"**.

In one embodiment, each of base/delta source files **110"/112"/115"** is identified with a universally unique identifier (UUID), as disclosed in co-pending U.S. patent application number <to be inserted>, filed contemporaneously, entitled "User Centric Source Control", which is hereby fully incorporated by reference (except for the reciprocating incorporation by reference). Each of the UUIDs universally identifies the particular base/delta source file **110"/112"/115"** among other base/delta source files of the program product as well as among other base/delta source files of all other program products of all other software vendors. For this embodiment, the versioning control information **116"** includes predecessor UUID information and other control information, such as locking and privileges, for the base/delta source files **110"/112"/115"**, as described in the co-pending application. As described earlier, sender **106"** provides these predecessor UUID and other control information to receiving system **140"**.

Also in like manner, receiver **146"** updates symbol tables **114"** stored in library **142"**, whenever it receives new operand substituting token to operand mappings from sending system **100"**. Receiver **146"** also stores versioning control information in library **140"**, upon receiving them from sending system **100"**, and making the versioning control information available for use on receiving system **140"**.

While the present invention is being described with **Fig. 1c** as an extension of **Fig. 1b**, those skilled in the art will appreciate that the present invention may also be practiced with **Fig. 1a** being extended with the additional base/delta and versioning control features of **Fig. 1c**, but without the additional tokenizing operand feature of **Fig. 1b**. In fact, those skilled in the art will appreciate that the present invention may be practiced with other additional features, and/or without some of the earlier described features.

Referring now to **Figures 3a-3b**, wherein one embodiment each of the operational steps of sending and receiving systems **100** and **140** are shown. In the remaining descriptions, when reference is made to an element, such as tokenizer **102**, unless specifically noted, the reference is intended to include all embodiments earlier described, i.e. tokenizer **102**, **102'** as well as **102"** of **Fig. 1a**, **1b** and **1c**. First, over in sending system **100**, as illustrated in **Fig. 3a**, at step **302**, tokenizer **102** is initially employed to transform the source files from the original form to the tokenized form, with at least the language elements being substituted by corresponding tokens. Step **304** is an optional step for those embodiments where at step **302**, operands are also substituted by corresponding tokens. Where applicable, tokenizer **102** further creates a symbol table, or new operand substituting token mapping entries for an existing symbol table, depending on whether the source file being processed is a first of a collection of interrelated source files or merely additional ones of the collection. Steps **302** and **304** are presented as two separate discrete steps for ease of understanding. They may be practiced as separate steps as described or as a single combined step.

Upon generating the transformed source files, sending system **100** awaits for requests for the source files, step **306**. As described earlier, the requests may be made by a local requestor, such as an application on sending system **100**, or by a remote requestor, such as an application on receiving system **140**. In any event, upon receipt of a request to provide selected ones

of the source files to receiving system **140**, sender **106** provides requested ones of the source files in the tokenized form, reducing transmission bandwidth requirement on communication medium **180**. Step **310** is also an optional step for those embodiments where at step **302**, operands are also substituted by tokens, and/or the source files are being kept in base/delta form with versioning control information. Where applicable, sender **106** further provides the symbol table, update entries for the symbol table, or versioning control information, as the situation may call for. Similarly, steps **308** and **310** are presented as two discrete steps for ease of understanding. They too may be practiced as separate steps as described or as one single combined step.

Upon providing the requested ones of source files, and other applicable symbol table(s) and/or versioning control information to receiving system **140**, sending system **100** returns to step **306**, unless sending system **100** is to terminate operation. Steps **308-310** are repeated as many times as necessary to satisfy the various requests received by sending system **100** for receiving system **140** and the likes. Furthermore, while for ease of understanding, **Fig. 3a** illustrates the process of tokenizing the source files as being performed for a number of source files before requests for selected ones of the source files are received and serviced, those skill in the art will appreciate that in alternate embodiments, the process of tokenizing the source files may be dynamically performed subsequent to receiving a request for the source files instead.

Over in receiving system **140**, as illustrated in **Fig. 3b**, at step **322**, receiving system **140** either proceeds with steps **324-326** or step **328** depending on whether it is receiver **146** who has received source files provided by sending system **100** or it is de-tokenizer **144** who has received a request to restore selected ones of the tokenized source files. If it is the former, receiver **146** stores the received source files in tokenized form into library **142** as described earlier. Step **326** is an optional step for those embodiments where

the operands are also substituted by tokens and/or the source files are being kept in base/delta form with versioning control information. Where applicable, receiver **146** also stores the symbol table or versioning control information into library **142** or updates the symbol table, as the situation may call for.

At step **328**, de-tokenizer **144** restores the source files from the tokenized form back to the original form, restoring at least the language element substituting tokens to the corresponding tokens. For embodiments where operands are also substituted by tokens, de-tokenizer **144** further restores the operand substituting tokens to the corresponding operands, using the appropriate symbol tables.

Upon responding to the receipt of source files in tokenized form or their associated information, or responding to requests to restore selected ones of the source files, receiving system **140** returns to step **322**, unless receiving system **140** is to terminate operation. Steps **324-326** and step **328** are repeated as many times as necessary to service the receipts and various requests received by receiving system **140**. Furthermore, while for ease of understanding, **Fig. 3b** illustrates the process of restoration as being performed "on-demand", those skill in the art will appreciate that in alternate embodiments, the process of restoration may also be performed in batch prior to making the restored source files available for use on receiving system **140**.

Referring now to **Figure 4**, wherein an exemplary application of the present invention to the provision of web pages by web server is illustrated. As shown, web site **400** and an exemplary client system **440** is coupled to one another through Internet **480**. Web site **400** provides web pages to client system **440** responsive to requests from client system **440**. Incorporated with the teachings of the present invention, web site **400** advantageously provides the requested web pages to client system **440** in the above described tokenized form, reducing the transmission bandwidth requirement on Internet

480, which as those skilled in the art will appreciate, will also likely to result in improving perceived response time to a user of client system 440.

As described earlier for sending system 100" of Fig. 1c, web site 400 includes library 402, tokenizer 404 and HTTP interface 406 (in the role of sender 106). Library 402 is used to store HTML web pages, JAVA scripts and so forth in original as well as tokenized form 410 and 412 (hereinafter simply web page or web pages), including symbol tables 414. For the illustrated embodiment, web pages 410 and 412 are kept in base/delta form having associated versioning control information 416. However, for preferred implementation reasons, versioning control information 416 are stored in a separate repository 418 as opposed to library 402. For alternate embodiments, repository 418 may be implemented as an integral part of library 402. Tokenizer 404 and HTTP interface 406 operate as described earlier for the corresponding elements of sending system 100" to effectuate the desired reduction in bandwidth requirement on Internet 480.

Similarly, as described earlier for receiving system 140" of Fig. 1c, client system 440 includes library 442, de-tokenizer 444 and browser 446 (in the role of receiver 146). Library 442 is used to store web pages in tokenized form 412 and symbol tables 414. Also for preferred implementation reasons, versioning control information 416 are stored in a separate repository 448. Likewise, for alternate embodiments, repository 448 may also be combined with library 442.

Browser 446 includes conventional elements found in many browsers known in the art, HTTP interface 462, HTML web page handler 464, JAVA™ and JavaScript execution engine 466, other script interpreter 468 (e.g. CGI), display interface 470, and a number of "plug-ins", shown as additional Active-X components 472. Included among these Active-X components 472 is a component that interacts with library 442 and repository 448 to store received

web pages **412**, symbol tables **414** and versioning control information **416** in library **442** and repository **448**, and selectively invoke detokenizer **414** to reconstitute and restore the web pages, as described earlier for corresponding elements of receiving system **140"**. Except for the teachings of the present invention incorporated in the particular Active-X component, all other elements perform their conventional functions known in the art, and their constitutions are well, accordingly, will not be further described. Additionally, those skilled in the art will also appreciate that the particular Active-X component may be provided integrally with the browser or complementarily as a supplemental function. In fact, the ability might be provided via other "extension" or "plug-in" technology. The browser may also be an integral function of an operating system having other conventional operating system functions such as a file subsystem, task scheduling and so forth.

While the above exemplary application is described in the context of the Internet and World Wide Web, those skilled in the art will appreciate that Internet **480** may be an internal private network of a corporation or an organization, with web site **400** and client system **460** being internal information servers and user computer systems respectively. Furthermore, as described in the incorporated by reference co-pending application, the provision of web pages in the above described base/delta form also has the advantage of enabling a user to selectively roll back to prior versions of the web pages. Those skilled in the art will also appreciate that for certain browsers known in the art, the described manner of provision also has the advantage of making it difficult for the unskilled users to determine the source content of the web pages.

Figure 5 illustrates one embodiment of an exemplary computer system suitable for use to practice the present invention, in particular as a user system. As shown, exemplary computer system **500** includes processor **502** and system memory **504** coupled to each other via system bus **506**. Coupled

also system bus **506** are non-volatile storage **508**, various user input/output devices **510** and communication interface **520**. Each of these elements perform its conventional functions known in the art. In particular, system memory **504** and non-volatile storage **508** are employed to store a working copy and a permanent copy of the programming instructions implementing the teachings of the present invention. The permanent copy of the programming instructions may be loaded into non-volatile storage **508** in the factory, or in the field, through distribution medium **522** or through communication interface **520**. As described earlier, any one of a number of recordable medium such as tapes and so forth may be employed. The constitution of these elements **502-520** are also well known, and accordingly will not be further described.

Thus, a novel method and apparatus for token based source file compression/decompression and its application has been described. While the present invention has been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

CLAIMS

What is claimed is:

1. In a computer system, a computer implemented source file transmission method comprising:

transforming a plurality of source files into a tokenized form, including substituting language elements present in the source files with corresponding tokens to reduce transmission bandwidth required to provide the source files to one or more other computer systems; and

providing the source files to the one or more other computer systems by transmitting the transformed source files in said tokenized form to the one or more other computer systems.

2. The method of claim 1, wherein the transformation further includes substituting operands of the source files with corresponding tokens and generating a symbol table with entries mapping the operand substituting tokens to the operands, and the provision further includes transmitting the symbol table to the one or more other computer systems.

3. The method of claim 1, wherein

the source files are in a base or a delta form,

the transformation further includes substituting operands with corresponding tokens, and generating a new entry for a symbol table for each new operand encountered and substituted by a new token; and

the provision further includes transmitting the new entries for the symbol table to the one or other computer systems.

4. The method of claim 1, wherein the source files are in either a bas or a delta form having associated versioning control information, and the provision

further includes transmitting the versioning control information to the one or more other computer systems.

5. The method of claim 4, wherein each of the base/delta source files are identified by an universally unique identifier (UUID), and the provision of the versioning control information to the one or more computer systems includes transmission of predecessor UUID information of the base/delta source files.

6. In a computer system, a computer implemented source file reception method comprising:

receiving a plurality of source files in a tokenized form from a second computer system, the source files being provided in the tokenized form to reduce transmission bandwidth requirement;

storing the source file in said tokenized form in the computer system; and

upon request, transforming requested ones of the stored source files back to an original non-tokenized form, including restoring language element substituting tokens of the source files to corresponding language elements.

7. The method of claim 6, wherein said receiving further includes receiving a symbol table having entries that map operand substituting tokens to operands, and the transformation further includes restoring operand substituting tokens of the source files to corresponding operands, using said received symbol table.

8. The method of claim 6, wherein

the source files are in either a base or a delta form,

said receiving further includes receiving from the second computer system new entries mapping new operand substituting tokens to new operands for a symbol table to which the base/delta source files are associated; and

the method further includes updating the symbol table with the received new entries.

9. The method of claim 6, wherein the source files are in either a base or a delta form having associated versioning control information, and said receiving further includes receiving the associated versioning control information from the second computer system.

10. The method of claim 9, wherein each of the base/delta source files is identified by an universally unique identifier (UUID), and the receiving of the associated versioning control information from the second computer system includes receiving predecessor UUID information of the base/delta source files.

11. A computer system comprising:

a tokenizer to transform a plurality of source files into a tokenized form, including substituting language elements present in the source files with corresponding tokens, to reduce transmission bandwidth required to provide the source files to one or more other computer systems; and

a transmitter having access to the transformed source files to transmit the transformed source files in said tokenized form to the one or more other computer systems.

12. The computer system of claim 11, wherein the tokenizer further substitutes operands of the source file with corresponding tokens, and generates a symbol table with entries mapping the operand substituting tokens to the operands, and the transmitter further transmits the symbol table to the one or more other computer systems.

13. The computer system of claim 11, wherein

the source files are in either a base or a delta form,

the tokenizer further substitutes operands with corresponding tokens, and generates a new entry for a symbol table for each new operand encountered and substituted by a new token; and

the transmitter further transmits the new entries for the symbol table to the one or more other computer systems.

14. The computer system of claim 11, wherein the source files are in either a base or a delta form having associated versioning control information, and the transmitter further transmits the versioning control information to the one or more other computer systems.

15. The computer system of claim 14, wherein each of the base/delta source files is identified by an universally unique identifier (UUID), and the transmitter transmits predecessor UUID information of the base/delta source files.

16. A computer system comprising:

a receiver to receive a plurality of source files in a tokenized form from a second computer system, the source files being provided in the tokenized form to reduce transmission bandwidth requirement;

a storage medium to store the source files in said tokenized form; and

a de-tokenizer, to selectively transform, upon request, requested ones of the stored source files back to an original non-tokenized form, including restoring language element substituting tokens of the requested ones of the source files to corresponding language elements.

17. The computer system of claim 16, wherein said receiver further receives a symbol table having entries that map operand substituting tokens to operands, and the de-tokenizer further restores operand substituting tok ns of th requested ones of the source files to corresponding operands, using said received symbol table.

18. The computer system of claim 16, wherein

the source files are in a base or a delta form, and

said receiver further receives from the second computer system new entries mapping new operand substituting tokens to new operands for a symbol table to which the base/delta source files are associated, and the receiver further updates the symbol table with the received new entries.

19. The computer system of claim 16, wherein the source files are in either a base or a delta form having associated versioning control information, and said receiver further receives the associated versioning control information from the second computer system.

20. The computer system of claim 19, wherein each of the base/delta source files is identified by an universally unique identifier (UUID), and the receiver receives predecessor UUID information of the base/delta source files.

21. An article of manufacture comprising:

a recordable medium having recorded thereon a plurality of programming instructions usable to program an apparatus to enable the apparatus to be able to transform a plurality of source files into a tokenized form, including substituting language elements present in the source files with corresponding tokens, to reduce transmission bandwidth required to provide the source files to one or more other apparatuses, and to enable the apparatus to transmit the transformed source files in said tokenized form to the one or more other computer systems.

22. The article of claim 21, wherein the programming instructions further enable the apparatus to substitute operands of the source files with corresponding tokens, generate a symbol table with entries mapping the

operand substituting tokens to the operands, and transmit the symbol table to the one or more other computer systems.

23. The article of claim 21, wherein the programming instructions further enable the apparatus to accommodate the source files being in either a base or a delta form, the programming instructions further enabling the apparatus to substitute operands with corresponding tokens, generate a new entry for a symbol table for each new operand encountered and substituted by a new token, as well as transmit the new entries for the symbol table to the one or more other computer systems.

24. An article of manufacture comprising:

a recordable medium having recorded thereon a plurality of programming instructions useable to program an apparatus to enable the apparatus to be able to receive a plurality of source files in a tokenized form from a second computer system, the source files being provided in the tokenized form to reduce transmission bandwidth requirement, to store the source files in said tokenized form, and to transform, upon request, requested ones of the stored source files back to an original non-tokenized form, including restoring language element substituting tokens of the requested ones of the source files to corresponding language elements.

25. The article of claim 24, wherein the programming instructions further enable the apparatus to be able to receive a symbol table having entries that map operand substituting tokens to operands, and to restore operand substituting tokens of the source files to corresponding operands, using said received symbol table.

26. The article of claim 24, wherein the programming instructions further enable the apparatus to accommodate the source files being in a base or a delta form, to receive from the second computer system new entries mapping

new operand substituting tokens to new operands for a symbol table to which the delta source file is associated, and to update the symbol table with the received new entries.

27. In a web server, a method comprising:

storing a plurality of web pages in a tokenized form with at least language elements present in the web pages in an original form substituted with corresponding tokens; and

upon request from a client computer system, providing requested ones of the stored web pages in the tokenized form to the client computer system.

28. The method of claim 27, wherein the web pages further having operands present in the web pages in the original form substituted with corresponding tokens, and entries mapping the operand substituting tokens to the operands being maintained in a symbol table, and the method further includes transmitting the symbol table to the client computer system.

29. The method of claim 27, wherein

the web pages are also in a base or a delta form, having operands present in the original form substituted with corresponding tokens, and having a new entry created in a symbol table for each new operand encountered and substituted by a new token, and

the method further includes transmitting the new entries for the symbol table to the client computer system.

30. A web server comprising:

a storage medium having stored therein a plurality of web pages that have been transformed into a tokenized form, with at least language elements present in the web pages in an original form having been substituted by corresponding tokens;

an interface to receive a request for one of the web pages from a client computer system; and

a transmitter to transmit the request web page, in the tokenized form, to the requesting client computer system.

31. The web server of claim 30, wherein the web pages further having operands present in the web pages in the original form substituted with corresponding tokens, and entries mapping the operand substituting tokens to the operands being maintained in a symbol table, and the transmitter further transmits the symbol table to the client computer system.

32. The web server of claim 30, wherein

the web pages are also in a base or a delta form, having operands present in the original form substituted with corresponding tokens, and having new entries created for a symbol table for each new operand encountered and substituted by a new token, and the transmitter further transmits the new entries for the symbol table to the client computer system.

33. In a computer system, a method comprising:

receiving a web page in a tokenized form from a web server; and

automatically transforming the web page back to an original form, including restoring language element substituting tokens back to corresponding language elements.

34. The method of claim 33, wherein said receiving further includes receiving a symbol table having entries that map operand substituting tokens to operands, and the transformation further includes restoring operand substituting tokens of the web page to corresponding operands, using said received symbol table.

35. The method of claim 33, wherein

the web page file is also in either a base or a delta form,

said receiving further includes receiving from the web server new entries mapping new operand substituting tokens to new operands for a symbol table to which the base/delta web page is associated; and

the method further includes updating the symbol table with the received new entries.

36. A browser comprising:

a first component to receive a web page in a tokenized form from a web server; and

a second component to automatically transform the web page back to an original form, including restoring language element substituting tokens back to corresponding language elements.

37. The browser of claim 36, wherein said first component further receives a symbol table having entries that map operand substituting tokens to operands, and said second component further restores operand substituting tokens of the web page to corresponding operands, using said received symbol table.

38. The browser of claim 36, wherein

the web page file is also in either a base or a delta form,

said first component further receives from the web server new entries mapping new operand substituting tokens to new operands for a symbol table to which the delta source file is associated, and updates the symbol table with the received new entries.

39. An operating system comprising:

a file subsystem; and

a browser having a first component to receive a web page in a tokenized form from a web server, and a second component to automatically transform the web page back to an original form, including restoring language element substituting tokens back to corresponding language elements.

40. The operating system of claim 39, wherein said first component of the browser further receives a symbol table having entries that map operand substituting tokens to operands, and said second component further restores operand substituting tokens of the web page to corresponding operands, using said received symbol table.

41. The operating system of claim 39, wherein

the web page file is also in either a base or a delta form,

said first component of the browser further receives from the web server new entries mapping new operand substituting tokens to new operands for a symbol table to which the delta source file is associated, and updates the symbol table with the received new entries.

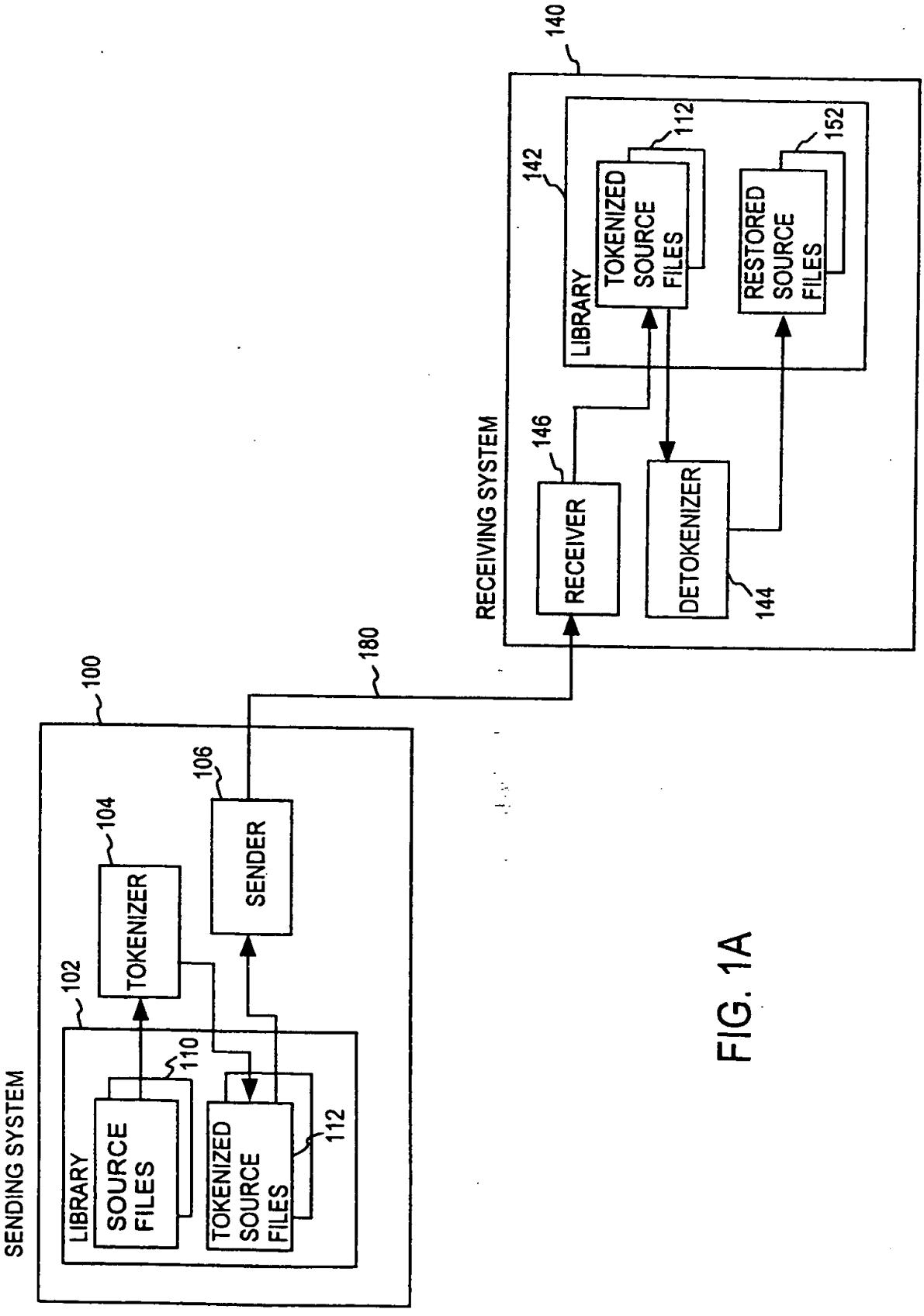

SENDING SYSTEM
100
102
LIBRARY
SOURCE FILES
110
TOKENIZED SOURCE FILES
112
TOKENIZER
104
SENDER
106
180
RECEIVING SYSTEM
140
142
LIBRARY
TOKENIZED SOURCE FILES
112
RESTORED SOURCE FILES
152
RECEIVER
146
DETOKENIZER
144

FIG. 1A

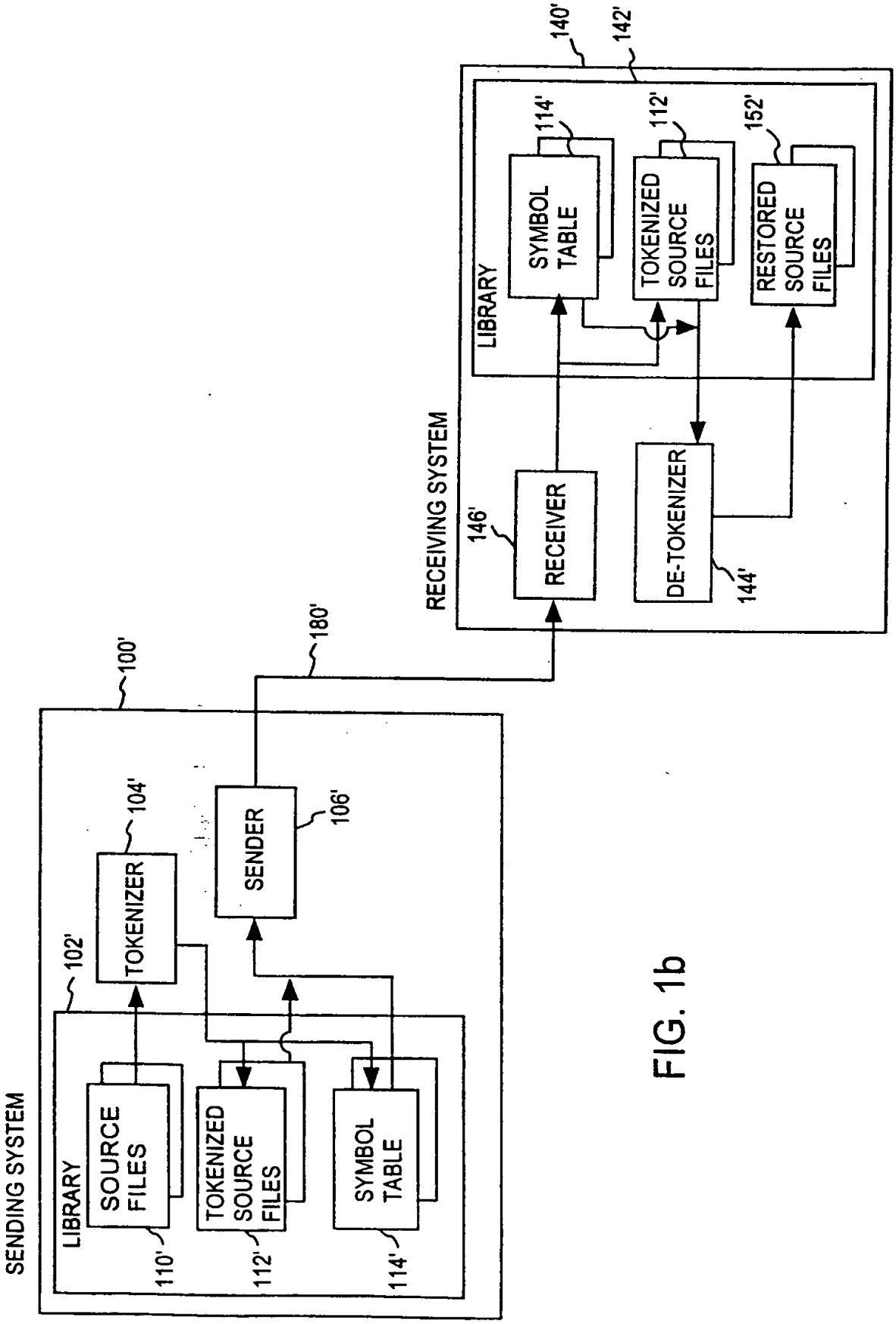

SENDING SYSTEM
100'
LIBRARY
102'
SOURCE FILES
110'
TOKENIZED SOURCE FILES
112'
SYMBOL TABLE
114'
TOKENIZER
104'
SENDER
106'
180'
RECEIVING SYSTEM
140'
142'
LIBRARY
SYMBOL TABLE
114'
TOKENIZED SOURCE FILES
112'
RESTORED SOURCE FILES
152'
RECEIVER
146'
DE-TOKENIZER
144'

FIG. 1b

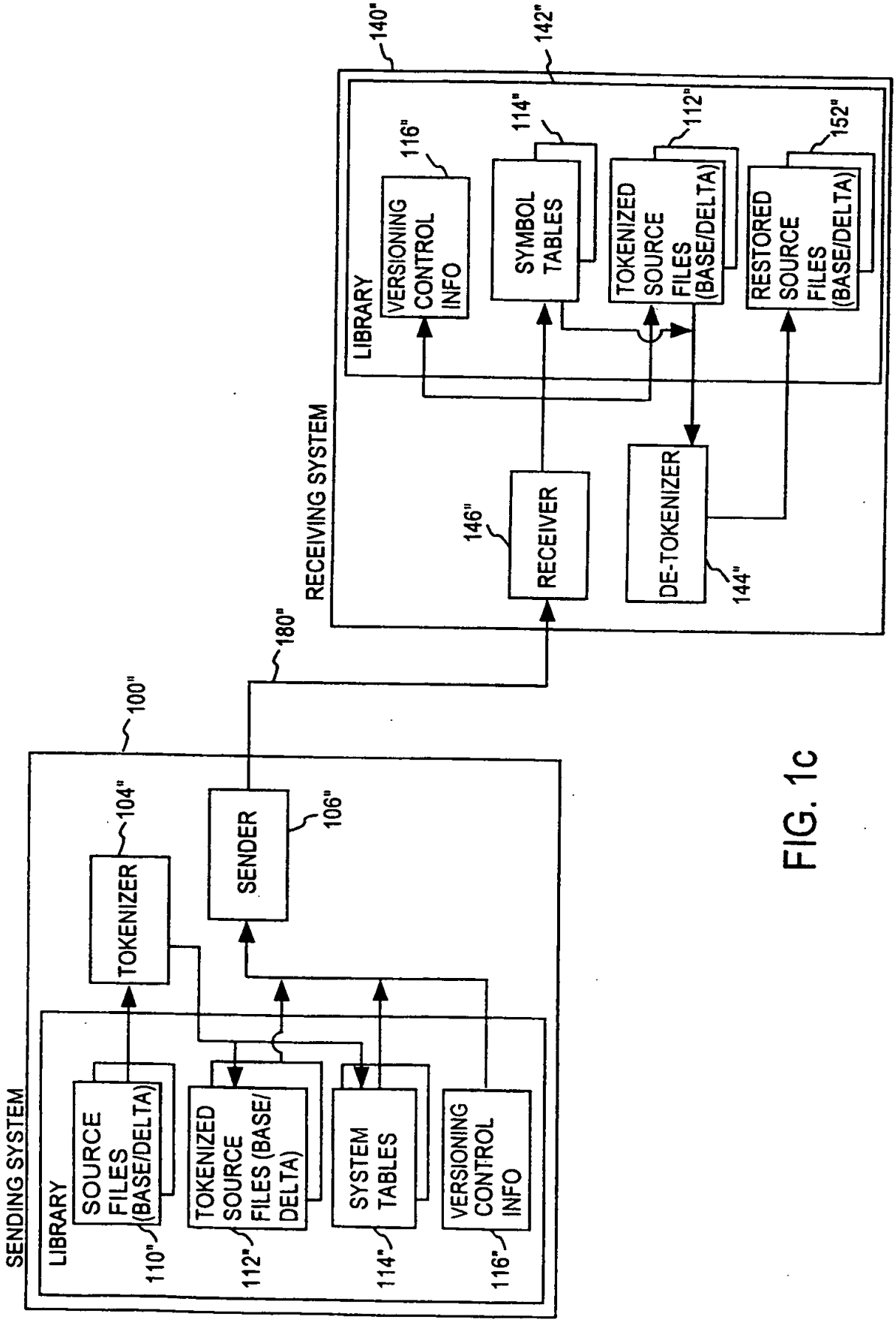

SENDING SYSTEM
100"
LIBRARY
SOURCE FILES (BASE/DELTA)
110"
TOKENIZED SOURCE FILES (BASE/ DELTA)
112"
SYSTEM TABLES
114"
VERSIONING CONTROL INFO
116"
TOKENIZER
104"
SENDER
106"
180"
RECEIVING SYSTEM
140"
LIBRARY
VERSIONING CONTROL INFO
116"
SYMBOL TABLES
114"
142"
TOKENIZED SOURCE FILES (BASE/DELTA)
112"
RESTORED SOURCE FILES (BASE/DELTA)
152"
RECEIVER
146"
DE-TOKENIZER
144"

FIG. 1c

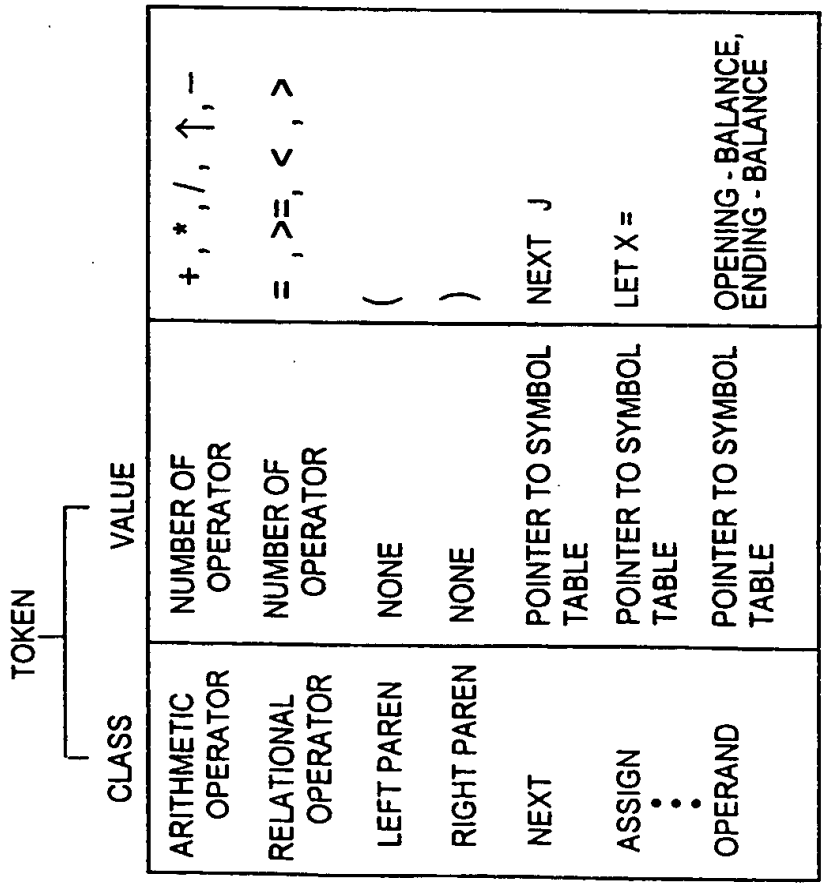

TOKEN
CLASS
VALUE
ARITHMETIC OPERATOR
NUMBER OF OPERATOR
+ , * , / , ↑, −
RELATIONAL OPERATOR
NUMBER OF OPERATOR
= , >=, < , >
LEFT PAREN
NONE
(
RIGHT PAREN
NONE
)
NEXT
POINTER TO SYMBOL TABLE
NEXT J
ASSIGN
POINTER TO SYMBOL TABLE
LET X =
OPERAND
POINTER TO SYMBOL TABLE
OPENING - BALANCE, ENDING - BALANCE

FIG. 2a

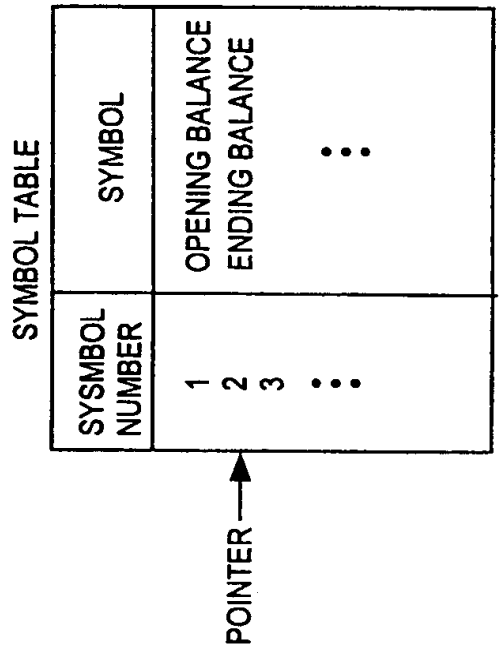

SYMBOL TABLE
SYSMBOL NUMBER
SYMBOL
1
2
3
OPENING BALANCE
ENDING BALANCE
POINTER

FIG. 2b

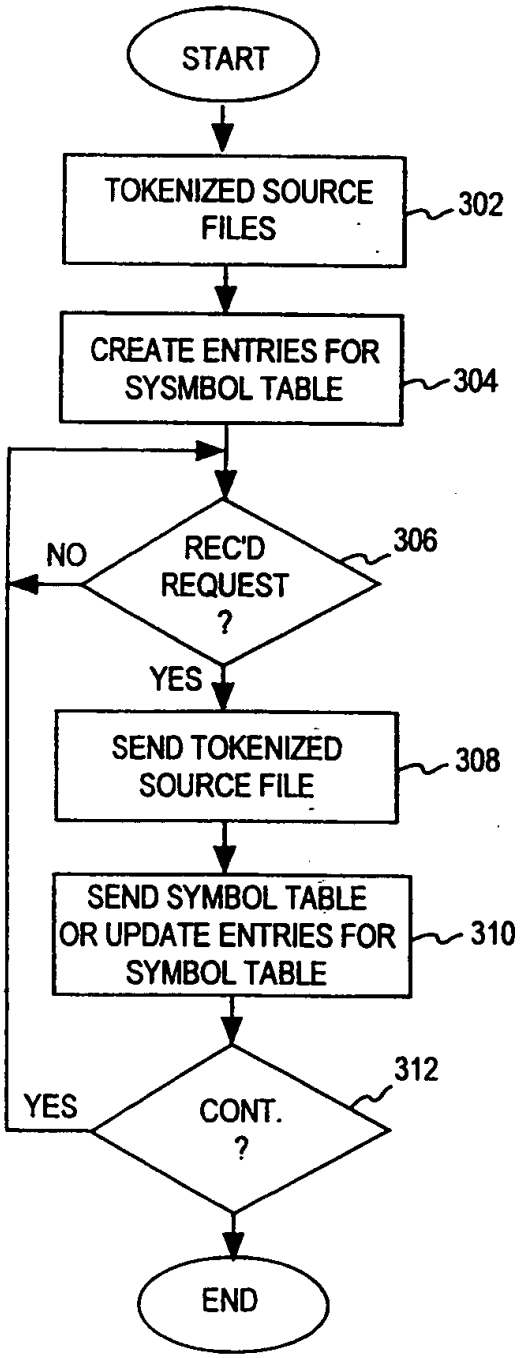

START
TOKENIZED SOURCE FILES
302
CREATE ENTRIES FOR SYSMBOL TABLE
304
REC'D REQUEST ?
306
NO
YES
SEND TOKENIZED SOURCE FILE
308
SEND SYMBOL TABLE OR UPDATE ENTRIES FOR SYMBOL TABLE
310
CONT. ?
312
YES
END

FIG. 3a

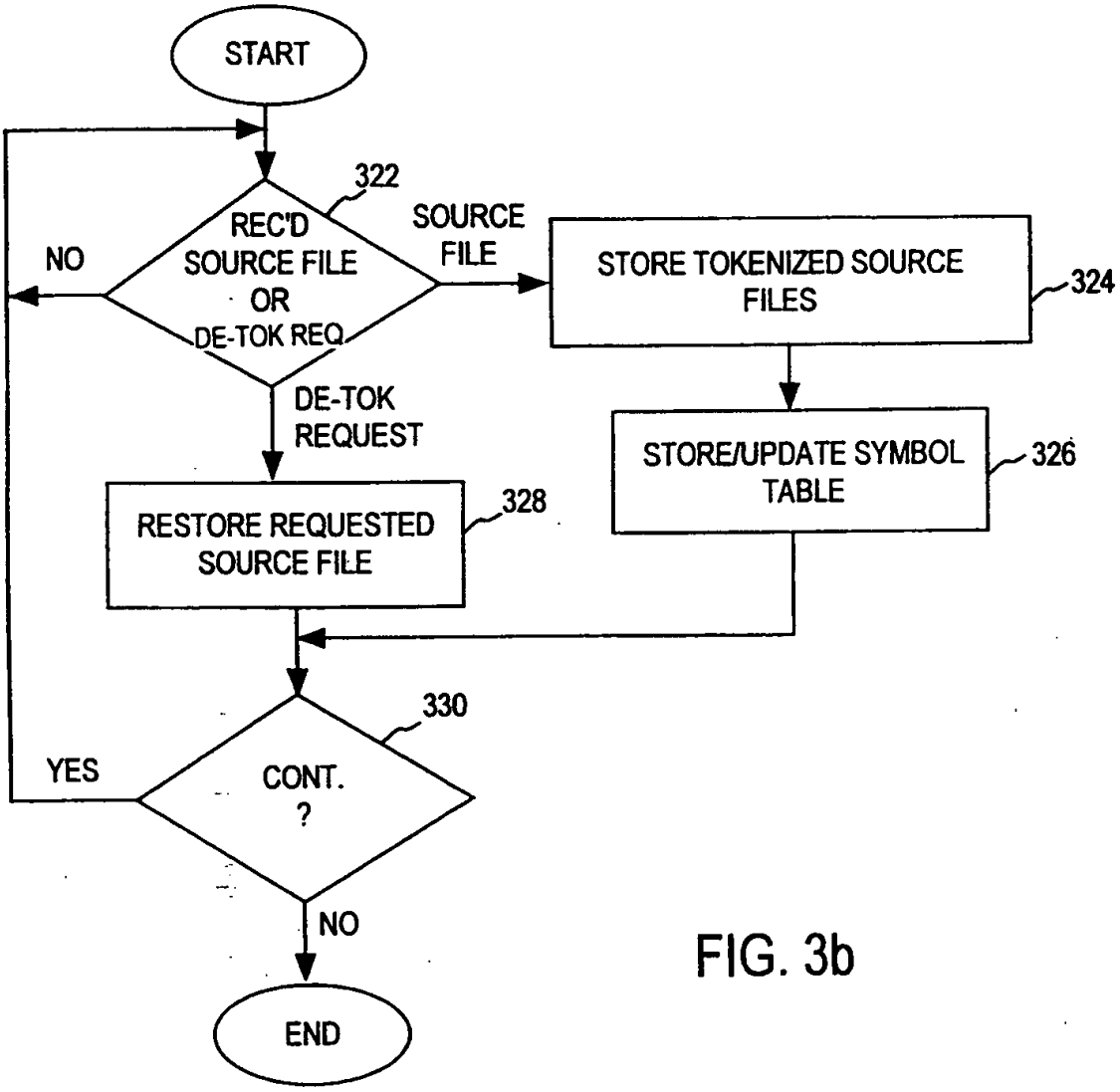

START
322
REC'D SOURCE FILE OR DE-TOK REQ
NO
SOURCE FILE
STORE TOKENIZED SOURCE FILES
324
DE-TOK REQUEST
STORE/UPDATE SYMBOL TABLE
326
RESTORE REQUESTED SOURCE FILE
328
330
YES
CONT. ?
NO
END

FIG. 3b

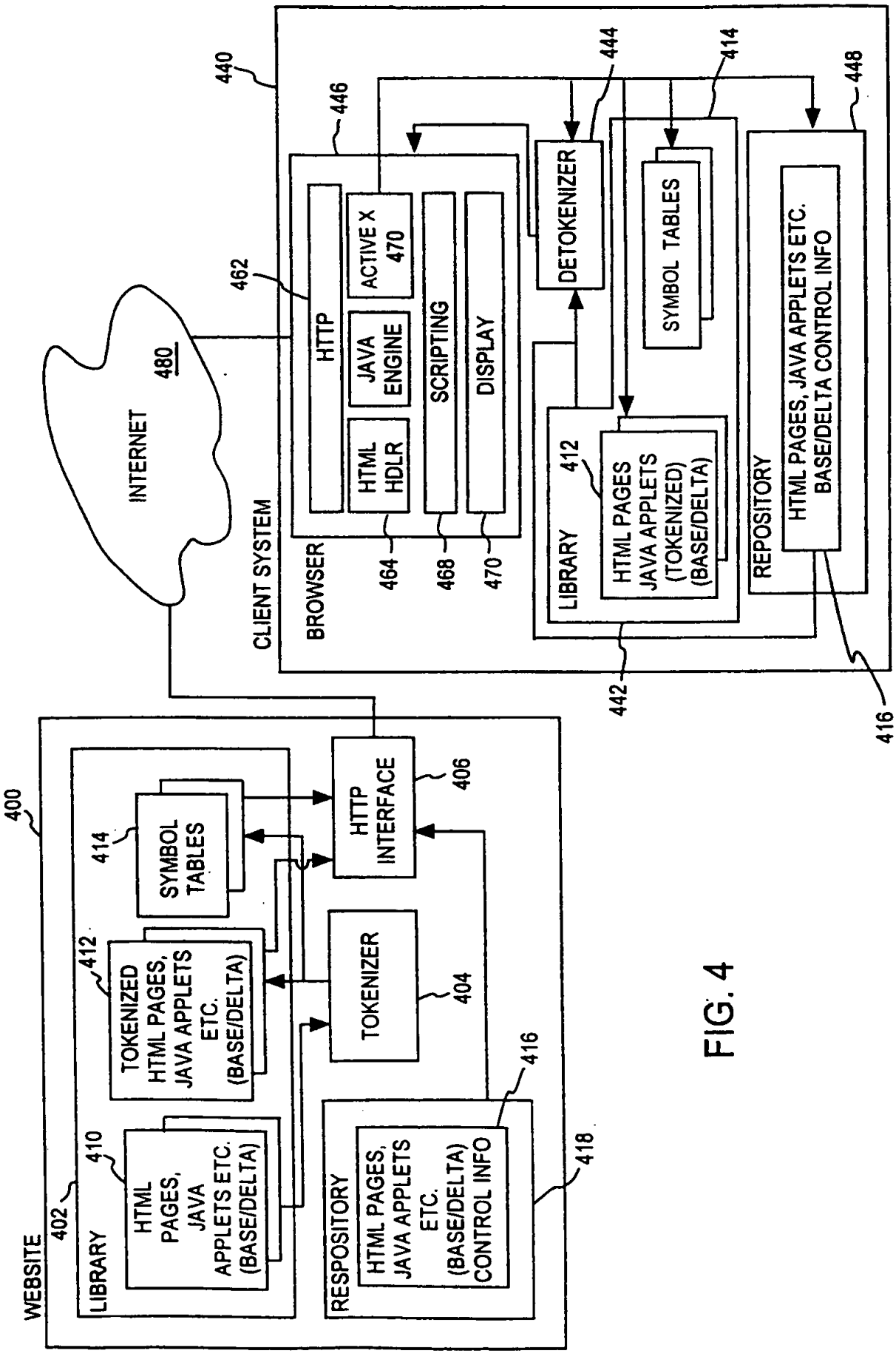

WEBSITE
400
LIBRARY
402
HTML PAGES, JAVA APPLETS ETC. (BASE/DELTA)
410
TOKENIZED HTML PAGES, JAVA APPLETS ETC. (BASE/DELTA)
412
SYMBOL TABLES
414
RESPOSITORY
HTML PAGES, JAVA APPLETS ETC. (BASE/DELTA) CONTROL INFO
416
418
TOKENIZER
404
HTTP INTERFACE
406
INTERNET
480
462
CLIENT SYSTEM
440
BROWSER
446
HTTP
HTML HDLR
JAVA ENGINE
ACTIVE X 470
SCRIPTING
DISPLAY
464
468
470
DETOKENIZER
444
LIBRARY
412
HTML PAGES JAVA APPLETS (TOKENIZED) (BASE/DELTA)
SYMBOL TABLES
414
REPOSITORY
448
HTML PAGES, JAVA APPLETS ETC. BASE/DELTA CONTROL INFO
442
416

FIG. 4

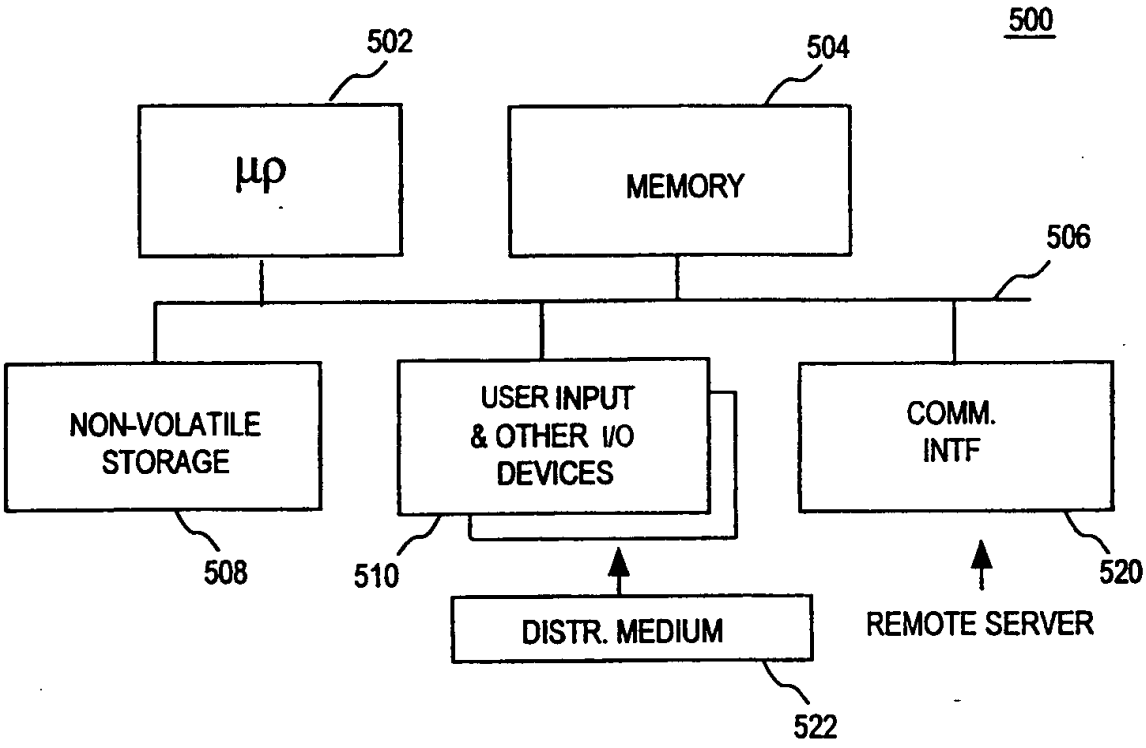

500
502
504
μp
MEMORY
506
NON-VOLATILE STORAGE
USER INPUT & OTHER I/O DEVICES
COMM. INTF
508
510
520
DISTR. MEDIUM
REMOTE SERVER
522

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/24919

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 15/16, 7/00, 13/38
US CL : 709/247; 710/68; 707/101
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/246, 247; 710/68; 707/101, 540

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

USPAT, Derwent, EPO, JPO, IEEE, Elsevier
search terms: differencing, token, compression, delta file, difference file, dictionary, parsing

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,E ---- Y,E	US 5,991,713 A (UNGER et al.) 23 November 1999, figs. 8, 12, and 13, col. 8 line 54 to col. 16 line 17.	1-2, 6-7, 11-12, 16-17, 21-22, 24-25, 27-28, 30-31, 33-34, 36-37, 39-40 ---------- 3-5, 8-10, 13-15, 18-20, 23, 26, 29, 32, 35, 38, 41

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 24 JANUARY 2000	Date of mailing of the international search report 17 FEB 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer Ahmad Matar

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/24919

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P ---- Y,P	US 5,884,014 A (HUTTENLOCHER et al.) 16 March 1999, figs. 4, 5, 9, and 10, col. 1 line 13 to col. 2 line 64, col. 26 line 52 to col. 28 line 51.	1-2, 6-7, 11-12, 16-17, 21-22, 24-25, 27-28, 30-31, 33-34, 36-37, 39-40 ---------- 3-5, 8-10, 13-15, 18-20, 23, 26, 29, 32, 35, 38, 41
X ---- Y	US 5,530,645 A (CHU) 25 June 1996, col. 2 line 57 to col. 5 line 20.	1-2, 6-7, 11-12, 16-17, 21-22, 24-25 ---------- 3-5, 8-10, 13-15, 18-20, 23, 26
Y	BLACK, A., et al., A Compact Representation for File Versions: A Preliminary Report, Proceedings of the Fifth Int'l. Conf. on Data Engineering, IEEE, May 1989, pages 321-329, particularly pages 321-322.	3-5, 8-10, 13-15, 18-20, 23, 26, 29, 32, 35, 38, 41
A	BELL, T., et al., Modeling for Text Compression, ACM Computing Surveys, December 1989, Vol. 21, No. 4, pages 557-591.	1-41